YAMAHA
RCX プロバイダ

Version 1.0.0

ユーザーズ ガイド

October 31, 2012

備考:

【改版履歴】

バージョン	日付	内容
1.0.0	2012/10/31	初版.

【対応機器】

機種	バージョン	注意事項

【ご注意】

本プロバイダを使用する場合は別途“RCX プロバイダ”ライセンスが必要です.

目次

1. はじめに 6
1.1. 非常停止スイッチの設置 6
2. プロバイダの概要 7
2.1. 概要 7
2.2. メソッド・プロパティ 8
2.2.1. CaoWorkspace::AddControllerメソッド 8
2.2.1.1. Connオプション 8
2.2.2. CaoController::AddRobotメソッド 9
2.2.3. CaoController::AddVariableメソッド 9
2.2.4. CaoController::Executeメソッド 10
2.2.5. CaoRobot::AddVariableメソッド 10
2.2.6. CaoRobot::Accelerateメソッド 10
2.2.7. CaoRobot::Changeメソッド 11
2.2.8. CaoRobot::Haltメソッド 11
2.2.9. CaoRobot::Moveメソッド 11
2.2.10. CaoRobot::Speedメソッド 12
2.2.11. CaoRobot::Executeメソッド 12
2.2.12. CaoRobot::get_IDメソッド 12
2.2.13. CaoRobot::put_IDメソッド 12
2.2.14. CaoVariable::get_Valueメソッド 13
2.2.15. CaoVariable::put_Valueメソッド 13
2.3. 変数一覧 13
2.3.1. コントローラクラス 13
2.3.2. ロボットクラス 14
2.4. エラーコード 15
3. コマンドリファレンス 17
3.1. コントローラクラス 17
3.1.1. CaoController::Execute(“EMGRST”) コマンド 17
3.1.2. CaoController::Execute(“ArithmeticExpression”) コマンド 17
3.1.3. CaoController::Execute(“StringExpression”) コマンド 18
3.1.4. CaoController::Execute(“PointExpression”) コマンド 18
3.1.5. CaoController::Execute(“ShiftExpression”) コマンド 19

3.1.6. CaoController::Execute(“SendControlCode”) コマンド.....19
3.1.7. CaoController::Execute(“NativeSend”) コマンド.....19
3.1.8. CaoController::Execute(“NativeReceive”) コマンド.....20
3.2. ロボットクラス.....21
3.2.1. CaoRobot::Execute(“ABSRST”) コマンド.....21
3.2.2. CaoRobot::Execute(“DRIVE”) コマンド.....22
3.2.3. CaoRobot::Execute(“DRIVEI”) コマンド.....22
3.2.4. CaoRobot::Execute(“ORIGIN”) コマンド.....23
3.2.5. CaoRobot::Execute(“PMOVE”) コマンド.....23
3.2.6. CaoRobot::Execute(“SERVO”) コマンド.....23
3.2.7. CaoRobot::Execute(“HAND”) コマンド.....24
3.2.8. CaoRobot::Execute(“RIGHTY”) コマンド.....24
3.2.9. CaoRobot::Execute(“LEFTY”) コマンド.....25
3.2.10. CaoRobot::Execute(“SHIFT”) コマンド.....25
3.2.11. CaoRobot::Execute(“ARCH”) コマンド.....25
3.2.12. CaoRobot::Execute(“ASPEED”) コマンド.....26
3.2.13. CaoRobot::Execute(“AXWGHT”) コマンド.....26
3.2.14. CaoRobot::Execute(“ORGORD”) コマンド.....27
3.2.15. CaoRobot::Execute(“OUTPOS”) コマンド.....27
3.2.16. CaoRobot::Execute(“PDEF”) コマンド.....27
3.2.17. CaoRobot::Execute(“TOLE”) コマンド.....28
3.2.18. CaoRobot::Execute(“WEIGHT”) コマンド.....28
3.2.19. CaoRobot::Execute(“TORQUE”) コマンド.....29
3.2.20. CaoRobot::Execute(“TRQTIME”) コマンド.....29

付録A. RCXコマンド対応表..... 30

Appendix A.1. コントローラクラス.....30
Appendix A.2. ロボットクラス.....31

1. はじめに

本書は YAMAHA ロボット RCX シリーズ用の CAO プロバイダのユーザーズガイドです. 本書で扱う CAO プロバイダ(CaoProvRCX.dll)を RCX プロバイダと呼びます.

次章に RCX プロバイダの概要, 3 章にコマンドリファレンスを記載しています.

1.1. 非常停止スイッチの設置

ロボットコントローラを使用になる前に, 非常の際にただちにロボットの運転を停止できるよう, 作業者が容易に操作できる位置に非常停止スイッチを設置してください.

(1) 非常停止スイッチは, 赤色にしてください.

(2) 非常停止の機能は, 作動させたあと自動的に復帰せず, また他の作業者が不用意に復帰させることができないようにしてください.

(3) 非常停止スイッチは, 電源スイッチとは別個に設けてください.

2. プロバイダの概要

2.1. 概要

RCX プロバイダは，YAMAHA ロボットコントローラに依存する部分を吸収し CAO プロバイダ・インターフェース仕様で規定された機能を提供する CAO プロバイダです．そのファイル形式は DLL(Dynamic Link Library)であり，CAO エンジンから使用時に動的にロードされます．RCX プロバイダを使用するにあたっては ORiN2SDK をインストールするか，下表を参照して手作業でレジストリ登録を行う必要があります．

表 A-1 RCX プロバイダ

ファイル名	CaoProvRCX.dll
ProgID	CaoProv.YAMAHA.RCX
レジストリ登録	regsvr32 CaoProvRCX.dll
レジストリ登録の抹消	regsvr32 /u CaoProvRCX.dll

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddControllerメソッド

RCX プロバイダでは AddController 時に通信用の接続パラメータを参照し，通信の接続を行います．

このときオプションで通信形態，タイムアウト，Ethernet 通信の場合は Telnet のユーザ名，パスワードを指定します．

書式　AddController(<bstrCtrlName:BSTR>,<bstrProvName:BSTR>,
<bstrPCName:BSTR>,<bstrOption:BSTR>)

<bstrCtrlName> : [in] ロボット名
<bstrProvName> : [in] オプション文字列
固定値 =" CaoProv.YAMAHA.RCX"
<bstrPcName> [in] プロバイダの実行マシン名
<bstrOption> [in] オプション文字列

以下にオプション文字列に指定するリストを示す．

表 A-1 CaoWorkspace::AddController のオプション文字列

オプション	意味
Conn=<接続パラメータ>	必須．通信形態とその接続パラメータを設定します．
[User=<ユーザ名>]	Ethernet 接続で RCX コントローラにログインするためのユーザ名を指定します．(デフォルト：admin)
[Password=<パスワード>]	Ethernet 接続で RCX コントローラにログインするためのパスワードを指定します．(デフォルト：無し)
[Timeout=<タイムアウト時間>]	送受信時のタイムアウト時間(ミリ秒)を指定します．(デフォルト：500)

2.2.1.1. Connオプション

以下に Conn オプションの接続パラメータ文字列を示します．ここで角括弧(“[]”)内のパラメータは省略可能を示します．また，各パラメータの解説中の下線部はオプション指定を省略した時のデフォルト値を示します．

・ **Ethernet デバイス**

“Conn=eth:<IP Address>[:<PortNo>]”

<IP Address>： 必須． 接続先の IP アドレス．
例 ：”127.0.0.1”, ”192.168.0.1”

<PortNo>： 接続先のポート番号．
例 ：“127.0.0.1:23”, ”192.168.0.1:5010”

・ **RS232C デバイス**

"Conn=com:[<ComPort>"[:<BaudRate>[:<Parity>:<DataBits>:<StopBits>]]]"

<ComPort> : COM ポート番号. '1'-COM1, '2'-COM2,...

<BaudRate> : 通信速度. 4800, 9600, 19200, 38400, 57600, 115200

<ByteSize> : パリティ. 'N'-NONE, 'E'-EVEN, 'O'-ODD

<DataBits> : データビット数. '7'-7bit, '8' – 8bit

<StopBits> : ストップビット数. '1'-1bit, '2'-2bit

2.2.2. CaoController::AddRobotメソッド

CaoProvRobot オブジェクトを生成します.

ここで指定するロボット名は任意の文字列で指定することができます. AddRobot メソッドを呼び出すと CaoRobot オブジェクトが取得できます.

書式 AddRobot(<bstrName:BSTR>[,<bstrOption:BSTR>])

< bstrName > : [in] ロボット名

< bstrOption > : [in] オプション文字列

表 A-1 CaoWorkspace::AddController のオプション文字列

オプション	意味
[SubRobot=<TRUE/FALSE>]	サブロボット用の RCX コマンドを使用するかどうかを設定します.(デフォルト:FALSE) この設定は CaoRobot::ID プロパティを使用して変更することができます.

2.2.3. CaoController::AddVariableメソッド

変数オブジェクトを生成します.

RCXプロバイダで実装されている変数は2.3.1を参照してください.

書式 AddVariable(<bstrVariableName:VT_BSTR>[,<vntOption:VT_BSTR>])

< bstrVariableName > : [in] 変数名

<bstrOption> : [in] オプション文字列

使用例

```
Dim aaa As Object
Dim bbb As Double
```

```
Set aaa = caoCtrl.AddVariable("@POS")
bbb = aaa.Value
```

2.2.4. CaoController::Executeメソッド

コマンドを実行します.

Execute メソッドの引数は, コマンドを BSTR,パラメータを VARIANT 配列で指定します.

各コマンドの詳細は3.1を参照してください.

書式 [<vntRet:VT_VARIANT>=]Execute(<bstrCmd:VT_BSTR>[,<vntParam:VT_VARIANT>])

< vntRet >	:	[out] コマンドの返り値
< bstrCmd >	:	[in] コマンド
< vntParam >	:	[in] パラメータ

2.2.5. CaoRobot::AddVariableメソッド

変数オブジェクトを生成します.

RCXプロバイダで実装されているシステム変数は2.3.2を参照してください.

書式 AddVariable(<bstrVariableName:VT_BSTR>[,<vntOption:VT_BSTR>])

< bstrVariableName >	:	[in] 変数名
<bstrOption>	:	[in] オプション文字列

2.2.6. CaoRobot::Accelerateメソッド

加速度及び減速度を設定します.

書式 Accelerate <lAxis:VT_I4>, <fAccel:VT_R4>, <fDecel:VT_R4>

< lAxis >	:	[in] 軸番号 指定した軸の加速度及び減速度を設定します. 軸番号に0を指定した場合は全ての軸に対して加速度及び減速度を設定します.
< fAccel >	:	[in] 加速度 加速度を設定します. 加速度に0を指定した場合は, 加速度の設定を行いません. このため減速度のみを設定したい場合は, この項目を0に設定してください.
< fDecel>	:	[in] 減速度 減速度を設定します.

減速度に0を指定した場合は，減速度の設定を行いません．このため加速度のみを設定したい場合は，この項目を0に設定してください．

2.2.7. CaoRobot::Changeメソッド

ハンドを変更します．

書式　Change <bstrName:VT_BSTR>

< bstrName >　：　[in] ハンド番号

2.2.8. CaoRobot::Haltメソッド

CaoRobotクラスのMove，Executeメソッド等で行うロボット動作命令を実行した場合，Haltメソッドによりロボット動作を途中で停止させることが出来ます．

書式　Halt <bstrOption:VT_BSTR>

< bstrOption >　：　[in] bstrOption（未使用）

2.2.9. CaoRobot::Moveメソッド

ロボットが指定した位置へ移動します．第1引数にMoveコマンドの種類，第2引数にVARIANT配列で移動先情報，速度を指定します．以下にMoveの仕様を示します．

書式　Move <lComp:VT_I4>,<vntParam : VT_VARIANT>, <bstrOpt:VT_BSTR>

< lComp >　：　[in] 補完指定 (VT_I4) 表 2-3 参照

このパラメータの内容は実行コマンドと補完指定を決定します．

1　：　MOVE P

2　：　MOVE L

3　：　MOVE C

4　：　MOVEI P

< vntParam >　：　[in] 移動先指定

移動先を文字列で指定します．

指定方法については「RCX シリーズ プログラミング説明書」を参照してください．

< bstrOpt>　：　[in] オプション

使用できるオプションについては「RCX シリーズ プログラミング説明書」を参照してください．

使用例

・ 単軸コントローラでの Move 使用例

```
Dim aaa As Object

Set aaa = caoCtrl.AddRobot("MF20")

caoRob.Move 1, "P13", "SPEED=50"   ' 1=PTP
caoRob.Move 2, "P11"                        ' 2=直線補間
caoRob.Move 3, "P10"                        ' 3=円弧補間
caoRob.Move 4, "P11", "S=50"                ' 現在位置から P11=170mm/deg 移動
```

2.2.10. CaoRobot::Speedメソッド

プログラム速度を変更します.

書式　Change <lAxis:VT_I4>, <fSpeed:VT_R4>

< fSpeed > : [in] 軸番号(未使用)

< fSpeed > : [in] 速度

2.2.11. CaoRobot::Executeメソッド

コマンドを実行します.

Execute メソッドの引数は, コマンドを BSTR,パラメータを VARIANT 配列で指定します.

各コマンドの詳細は3.2を参照してください.

書式　[<vntRet:VT_VARIANT>=]Execute(<bstrCmd:VT_BSTR>[,<vntParam:VT_VARIANT>])

< vntRet > : [out] コマンドの返り値

< bstrCmd > : [in] コマンド

< vntParam > : [in] パラメータ

2.2.12. CaoRobot::get_IDメソッド

ロボットオブジェクトの対象がメインロボットとサブロボットのどちらであるかを取得します.

取得するIDの値の内容については, 表 A-1を参照してください.

表 A-1 ID 値とロボット種別

ID	ロボット種別
1	メインロボット
2	サブロボット

2.2.13. CaoRobot::put_IDメソッド

メインロボットとサブロボットを切り替えます.

設定するIDの値の内容については，表 A-1を参照してください．

2.2.14. CaoVariable::get_Valueメソッド

変数の値を取得します．

取得する値の詳細については2.3を参照してください．

2.2.15. CaoVariable::put_Valueメソッド

変数の値を設定します．

設定する値の詳細については2.3を参照してください．

2.3. 変数一覧

2.3.1. コントローラクラス

表 A-1 コントローラクラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@Timeout	VT_I4	通信のタイムアウト時間 この値は，AddController()の Timeout オプションで指定することができます．	○	○
@CONFIG	VT_BSTR VT_ARRAY	コントローラ構成を取得します． 取得した情報は配列に以下の順番で格納されます． <設定名> <軸設定> <標準インターフェース> <オプション機器> <その他の設定>	○	-
@EXELVL	VT_BSTR	実行レベル状態	○	-
@MOD	VT_BSTR	モード状態	○	-
@MSG	VT_BSTR	メッセージ	○	-
@UNIT	VT_BSTR	ポイント単位座標系	○	-
@VER	VT_BSTR	バージョン	○	-

@MEM	VT_I4 VT_ARRAY	メモリ残容量 取得した情報は配列に以下の順番で格納されます. <ソース領域残容量> <オブジェクト領域残容量>	○	-
@EMG	VT_I4	非常停止状態 0 正常状態 1 非常停止状態	○	-
@SELFCHK	VT_BSTR VT_ARRAY	自己診断によるエラー状態 エラーがない場合は VT_EMPTY を返します.	○	-
@OPSLOT	VT_BSTR VT_ARRAY	オプションスロット状態を返します.	○	-

表 A-2 コントローラクラス ユーザ変数一覧

変数名	データ型	説明	属性	
			get	put
P	VT_I4 VT_ARRAY または VT_R4 VT_ARRAY	ポイントデータ 変数名の後ろにポイント番号を指定します. データの要素数は 6. データ型によって設定される単位系が異なります. VT_I4 | VT_ARRAY パルス単位 VT_R4 | VT_ARRAY ミリ単位	○	○
S	VT_I4 VT_ARRAY または VT_R4 VT_ARRAY	シフトデータ 変数名の後ろにシフト番号を指定します. データの要素数は 4. データ型によって設定される単位系が異なります. VT_I4 | VT_ARRAY パルス単位 VT_R4 | VT_ARRAY ミリ単位	○	○

2.3.2. ロボットクラス

表 A-1 ロボットクラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@ARM	VT_BSTR VT_ARRAY	アーム状態 取得した情報は配列に以下の順番で格納されます. <現在のアーム設定状態> <プログラムリセット時のアーム設定状態>	○	-
@ORIGIN	VT_BSTR	原点復帰状態	○	-
@ABSRST	VT_BSTR	アブソリュートリセット状態	○	-
@SERVO	VT_BSTR	サーボ状態	○	-
@SPEED	VT_BSTR VT_ARRAY	速度状態 取得した情報は配列に以下の順番で格納されます. <自動移動速度設定> <手動移動速度設定>	○	-
@WHERE	VT_I4 VT_ARRAY	メインロボットのパルス座標系現在位置	○	-
@WHERE2	VT_I4 VT_ARRAY	サブロボットのパルス座標系現在位置	○	-
@WHRXY	VT_R4 VT_ARRAY	メインロボットのミリ単位系現在位置	○	-
@WHRXY2	VT_R4 VT_ARRAY	サブロボットのミリ単位系現在位置	○	-
@SHIFT	VT_BSTR	シフト状態	○	-
@HAND	VT_BSTR	ハンド状態	○	-

2.4. エラーコード

RCXプロバイダでは，以下の固有エラーコードが定義されています．ORiN2 共通エラーについては，「ORiN2 プログラミングガイド」のエラーコードの章を参照してください．

表 2-2 独自エラーコード一覧

エラー名	エラー番号	説明
E_CAOP_NO_LICENSE	0x80100000	ライセンスがありません． 追加ライセンスを購入してください．

RCX コマンドエラー	0x8011xxyy	RCX のコマンド実行時にエラーが発生した場合は, を xx の箇所にエラーグループ番号, yy の箇所にエラー分類番号を入れて返します. エラーコードの内容については RCX のマニュアルを参照してください.

3. コマンドリファレンス

本章では CaoController::Execute,CaoRobot::Execute メソッドの各コマンドについて解説します. 各コマンドの詳細動作については YAMAHA ロボットコントローラ取扱説明書を参照してください.

3.1. コントローラクラス

表 A-1 CaoController::Execute コマンド一覧

コマンド	機能	
EMGRST	非常停止の解除	P. 17
ArithmeticExpression	数値式演算	P. 17
StringExpression	文字列式演算	P. 18
PointExpression	ポイント式演算	P. 18
ShiftExpression	シフト式演算	P. 19
SendControlCode	制御コマンドの送信	P. 19
NativeSend	データの送信	P. 19
NativeReceive	データの受信	P. 20

3.1.1. CaoController::Execute(“EMGRST”) コマンド

RCX コントローラの内部非常停止フラグを解除します.

書式　EMGRST()

戻り値　：　なし

使用例

```
caoCtrl.Execute(“EMGRST”) ‘ 非常停止を解除
```

3.1.2. CaoController::Execute(“ArithmeticExpression”) コマンド

指定した数値式の演算を行い, 結果を取得します.

数値式の指定方法については,「RCX シリーズ プログラミング説明書」を参照してください.

書式　ArithmeticExpression(<bstrExpression:VT_BSTR>)

< bstrExpression >　：　[in] 数値式(VT_VARIANT)

戻り値 : [out] 演算結果 (VT_R8)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.Execute("arithmeticExpression","SQR(100 * 5)")  ' 2.23606E01
```

3.1.3. CaoController::Execute("StringExpression") コマンド

指定した文字列式の演算を行い, 結果を取得します.
文字列式の指定方法については,「RCX シリーズ プログラミング説明書」を参照してください.

書式 StringExpression(<bstrExpression:VT_BSTR>)

< bstrExpression > : [in] 文字列式(VT_VARIANT)
戻り値 : [out] 演算結果 (VT_BSTR)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.Execute("StringExpression", """ABC"" + ""DEF""")  'ABCDEF
```

3.1.4. CaoController::Execute("PointExpression") コマンド

指定したポイント式の演算を行い, 結果を取得します.
ポイント式の指定方法については,「RCX シリーズ プログラミング説明書」を参照してください.

書式 PointExpression(<bstrExpression:VT_BSTR>)

< bstrExpression > : [in] ポイント式(VT_VARIANT)
戻り値 : [out] 演算結果
演算結果の単位系により取得するデータ型が異なります.
VT_I4 | VT_ARRAY パルス単位
VT_R4 | VT_ARRAY ミリ単位

使用例

```
Dim vRes As Variant
vRes = caoCtrl.Execute("PointExpression", "P1+WHRXY")
```

3.1.5. CaoController::Execute(“ShiftExpression”) コマンド

指定したシフト式の演算を行い，結果を取得します.
シフト式の指定方法については，「RCX シリーズ プログラミング説明書」を参照してください.

書式　ShiftExpression(<bstrExpression:VT_BSTR>)

< bstrExpression > : [in] シフト式(VT_VARIANT)
戻り値 : [out] 演算結果
演算結果の単位系により取得するデータ型が異なります.
VT_I4 | VT_ARRAY パルス単位
VT_R4 | VT_ARRAY ミリ単位

使用例

```
Dim vRes As Variant
vRes = caoCtrl.Execute("ShiftExpression", "S1")
```

3.1.6. CaoController::Execute(“SendControlCode”) コマンド

コントローラに対して，1 バイトの制御コードを送信します. 制御コードについては，YAMAHA ロボットコントローラ取扱説明書を参照してください.

書式　SendControlCode (<bytCode:VT_UI1>)

< bytCode > : [in] 制御コード (VT_UI1)

0x03	^C:ORG,XINC,XDEC などの中断

戻り値 : なし

使用例

```
caoCtrl.Execute(“SendControlCode”, &H03)     ` ^C 0x03 を送信
```

3.1.7. CaoController::Execute(“NativeSend”) コマンド

コントローラに対して，Telenet 通信によるデータを送信します. 送信コマンドとパラメータを文字列で指定してください.

書式　NativeSend (<bstrNativeText>)

< bstrNativeText > : [in] 送信データ (VT_BSTR)
戻り値 : なし

使用例

```
caoCtrl.Execute("NativeSend", "@?ALM 0,2")　　` @?ALM コマンドでパラメータ"0,2"を送信
```

3.1.8. CaoController::Execute("NativeReceive") コマンド

コントローラに対して, Telenet 通信によるデータを受信します. コントローラからの送信データを文字列で取得します. "OK c/r l/f", "NG c/r l/f"などは含まれません.

書式　NativeReceive ()

戻り値 : [out] コントローラからの送信データ (VT_BSTR)

使用例

```
Dim aaa As String
aaa = caoCtrl.Execute("NativeReceive")　　' コントローラからの送信データを受信
```

3.2. ロボットクラス

表 A-1 CaoRobot::Execute コマンド一覧

コマンド	機能	
ABSRST	アブソリュートモータ軸の原点復帰	P. 21
DRIVE	軸単位の絶対移動命令	P. 22
DRIVEI	軸単位の相対移動命令	P. 22
ORIGIN	インクリメンタル仕様の軸の原点復帰	P. 23
PMOVE	パレット移動命令	P. 23
SERVO	サーボ状態の設定	P. 23
HAND	ハンドの定義	P. 24
RIGHTY	右手系に設定	P. 24
LEFTY	左手系に設定	P. 25
SHIFT	シフト座標の設定	P. 25
ARCH	アーチ位置パラメータの変更	P. 25
ASPEED	自動移動速度の設定	P. 26
AXWGHT	軸先端質量の設定	P. 26
ORGORD	原点復帰処理順番の設定	P. 27
OUTPOS	アウト有効位置パラメータの設定	P. 27
PDEF	パレットの定義	P. 27
TOLE	公差パラメータの変更	P. 28
WEIGHT	先端質量パラメータの変更	P. 28
TORQUE	トルクの設定をします.	P. 29
TRQTIME	トルクタイムアウトの設定をします.	P. 29

3.2.1. CaoRobot::Execute(“ABSRST”) コマンド

ロボットのアブソリュートモータ軸の原点復帰動作を実行します.

書式　ABSRST()

戻り値　：　なし

使用例

```
Dim rbt As Object
Set rbt = caoCtrl.AddRobot(“MF20”)
rbt.Execute “ABSRST”　　‘ 原点復帰
```

3.2.2. CaoRobot::Execute("DRIVE") コマンド

軸単位の絶対位置移動命令を実行します.

位置指定, オプション文字列の指定方法については「RCX シリーズ プログラミング説明書」を参照してください.

書式　DRIVE(< vntPoints:VT_VARIANT | VT_ARRAY>[, <bstrOption:VT_BSTR>])

< vntPoints > : [in] 移動位置指定 (VARIANT | VT_ARRAY)

移動位置n	(VARIANT | VT_ARRAY)
軸番号	(VT_I4)
位置指定	(VT_BSTR)

<bstrOption> [in] オプション文字列 (VT_BSTR)

戻り値 : なし

使用例

```
Dim rbt As Object
Set rbt = caoCtrl.AddRobot("MF20")
rbt.Execute "Drive", array(array(array(1, "P11"), array(2, "P10")), "S=100")
```

3.2.3. CaoRobot::Execute("DRIVEI") コマンド

軸単位の相対位置移動命令を実行します.

位置指定, オプション文字列の指定方法については「RCX シリーズ プログラミング説明書」を参照してください.

書式　DRIVEI(< vntPoints:VT_VARIANT | VT_ARRAY>[, <bstrOption:VT_BSTR>])

< vntPoints > : [in] 移動位置指定 (VARIANT | VT_ARRAY)

移動位置n	(VARIANT | VT_ARRAY)
軸番号	(VT_I4)
位置指定	(VT_BSTR)

<bstrOption> [in] オプション文字列 (VT_BSTR)

戻り値 : なし

使用例

```
Dim rbt As Object
Set rbt = caoCtrl.AddRobot("MF20")
```

```
rbt.Execute "Drive1", array(array(array(1, "P11"), array(2, "P10")), "S=100")
```

3.2.4. CaoRobot::Execute("ORIGIN") コマンド

インクリメンタル仕様の軸の原点動作復帰を実行します.

書式　ORIGIN()

戻り値　：　なし

使用例

```
Dim rbt As Object
Set rbt = caoCtrl.AddRobot("MF20")
rbt.Execute "ORIGIN"　　' 原点復帰
```

3.2.5. CaoRobot::Execute("PMOVE") コマンド

パレット移動命令を実行します.

オプション文字列の指定方法については「RCX シリーズ プログラミング説明書」を参照してください.

書式　PMOVE(<lPalette:VT_I4>, <lPos:VT_I4>[, <bstrOption:VT_BSTR>])

< lPalette >	：	[in] パレット番号 (VT_I4)
< lPos >		[in] ポイント番号 (VT_I4)
< bstrOption >		[in] オプション文字列 (VT_BSTR)
戻り値	：	なし

使用例

```
Dim rbt As Object
Set rbt = caoCtrl.AddRobot("MF20")
rbt.Execute "PMove", Array(1, 5, "S=50")
```

3.2.6. CaoRobot::Execute("SERVO") コマンド

パレット移動命令を実行します.

軸番号を省略した場合は, 全軸指定となります.

書式　SERVO (<bstrState:VT_BSTR>[, <lAxis:VT_I4>])

< bstrState > : [in] サーボ状態 (VT_BSTR)
< lAxis > [in] 軸番号 (VT_I4)
戻り値 : なし

使用例

```
Dim rbt As Object
Set rbt = caoCtrl.AddRobot("MF20")
rbt.Execute "Servo", array("ON")
```

3.2.7. CaoRobot::Execute("HAND") コマンド

ハンドの定義を行います.

書式 HAND (<lNo:VT_I4>, <fPara1:VT_R4>, <fPara2:VT_R4>, <fPara3:VT_R4>, <fPara4:VT_R4>, <bROption:VT_BOOL>)

<lNo > : [in] ハンド番号 (VT_I4)
<fPara1> [in] パラメータ1 (VT_R4)
<fPara2> [in] パラメータ2 (VT_R4)
<fPara3> [in] パラメータ3 (VT_R4)
<fPara4> [in] パラメータ4 (VT_R4)
<bROption> [in] R オプション (VT_BOOL)
戻り値 : なし

使用例

```
Dim rbt As Object
Set rbt = caoCtrl.AddRobot("MF20")
rbt.Execute "Hand", array(1, 0, 50.0, 0, 0) ' Y 軸 50mm に HAND1 定義
```

3.2.8. CaoRobot::Execute("RIGHTY") コマンド

直交座標系で指定されたポイントへ右手系で移動するように設定します.

書式 RIGHTY ()

戻り値 : なし

使用例

```
Dim rbt As Object
Set rbt = caoCtrl.AddRobot("MF20")
rbt.Execute "RIGHTY"       ' 右手系に設定
```

3.2.9. CaoRobot::Execute("LEFTY") コマンド

直交座標系で指定されたポイントへ左手系で移動するように設定します.

書式　LEFTY ()

戻り値　：　なし

使用例

```
Dim rbt As Object
Set rbt = caoCtrl.AddRobot("MF20")
rbt.Execute "LEFTY"        '右手系に設定
```

3.2.10. CaoRobot::Execute("SHIFT") コマンド

シフト変数を指定し, そこで指定されるシフトデータでシフト座標を設定します.

書式　SHIFT (<bstrShift:VT_BSTR>)

<bstrShift>　：　[in] シフト指定　(VT_BSTR)

戻り値　：　なし

使用例

```
Dim rbt As Object
Set rbt = caoCtrl.AddRobot("MF20")
rbt.Execute "SHIFT", "S1"          ' S1 のシフト使用
```

3.2.11. CaoRobot::Execute("ARCH") コマンド

アーチ位置パラメータを設定します.

軸番号を省略した場合は, 全軸指定となります.

書式　ARCH (<lPos:VT_I4>[, <lAxis:VT_I4>])

< lPos >　：　[in] アーチ位置パラメータ (VT_I4)
< lAxis >　　[in] 軸番号 (VT_I4)
戻り値　：　なし

使用例

```
Dim rbt As Object
Set rbt = caoCtrl.AddRobot("MF20")
rbt.Execute "Arch", array(100000, 2)
```

3.2.12. CaoRobot::Execute("ASPEED") コマンド

自動移動速度を変更します.

書式　ASPEED (<lSpeed:VT_I4>)

< lSpeed >　：　[in] スピード (VT_I4)
戻り値　：　なし

使用例

```
Dim rbt As Object
Set rbt = caoCtrl.AddRobot("MF20")
rbt.Execute "ASPEED", 10
```

3.2.13. CaoRobot::Execute("AXWGHT") コマンド

軸先端質量パラメータを変更します.

書式　AXWGHT (<lAxis:VT_I4>, <lVal:VT_I4>)

< lAxis >　：　[in] 軸番号 (VT_I4)
< lVal >　：　[in] 先端質量 (VT_I4)
戻り値　：　なし

使用例

```
Dim rbt As Object
```

```
Set rbt = caoCtrl.AddRobot("MF20")
rbt.Execute "AxWght", array(2,5)
```

3.2.14. CaoRobot::Execute("ORGORD") コマンド

原点復帰動作及びアブソサーチ動作を行う軸順序パラメータを設定します.

書式　ORGORD (<lOrder:VT_I4>)

< lOrder >	:	[in] 軸順序パラメータ (VT_I4)
戻り値	:	なし

使用例

```
Dim rbt As Object
Set rbt = caoCtrl.AddRobot("MF20")
rbt.Execute "Orgord", 2
```

3.2.15. CaoRobot::Execute("OUTPOS") コマンド

アウト有効位置パラメータを変更します.

軸番号を省略した場合は, 全軸指定となります.

書式　OUTPOS (<lPos:VT_I4>[, <lAxis:VT_I4>])

< lPos >	:	[in] アウト有効位置 (VT_I4)
< lAxis >		[in] 軸番号 (VT_I4)
戻り値	:	なし

使用例

```
Dim rbt As Object
Set rbt = caoCtrl.AddRobot("MF20")
rbt.Execute "Outpos", array(100000, 2)
```

3.2.16. CaoRobot::Execute("PDEF") コマンド

パレット移動命令を実行するためのパレットを定義します.

書式　PDEF (<lNo:VT_I4>, <lP12:VT_I4>, <lP13:VT_I4>[, <lP15:VT_I4>])

< lNo > : [in] パレット番号 (VT_I4)
< lP12> [in] P1-P2 間の点数 (VT_I4)
< lP13> [in] P1-P3 間の点数 (VT_I4)
< lP15> [in] P1-P5 間の点数 (VT_I4)
戻り値 : なし

使用例

```
Dim rbt As Object
Set rbt = caoCtrl.AddRobot("MF20")
rbt.Execute "PDef", Array(1, 3, 4, 2)        ' パレット 3x4x2
```

3.2.17. CaoRobot::Execute("TOLE") コマンド

公差パラメータを変更します.

書式 TOLE (<lPos:VT_I4>[, <lAxis:VT_I4>])

< lPos > : [in] アウト有効位置 (VT_I4)
< lAxis > [in] 軸番号 (VT_I4)
戻り値 : なし

使用例

```
Dim rbt As Object
Set rbt = caoCtrl.AddRobot("MF20")
rbt.Execute "Tole", array(1)
```

3.2.18. CaoRobot::Execute("WEIGHT") コマンド

先端質量パラメータを変更します.

書式 WEIGHT (<lVal:VT_I4>)

< lOrder > : [in] 先端質量パラメータ (VT_I4)
戻り値 : なし

使用例

```
Dim rbt As Object
Set rbt = caoCtrl.AddRobot("MF20")
rbt.Execute "Weight", 20 '20kg
```

3.2.19. CaoRobot::Execute("TORQUE") コマンド

指定された軸の最大トルク指令値を変更します.

書式　TORQUE (<lAxis:VT_I4>, <lVal:VT_I4>)

< lAxis >	:	[in] 軸番号 (VT_I4)
< lVal >	:	[in] トルク指令値 (VT_I4)
戻り値	:	なし

使用例

```
Dim rbt As Object
Set rbt = caoCtrl.AddRobot("MF20")
rbt.Execute "Torque", array(2, 100)                    ' 2 軸のトルクを最大にする
```

3.2.20. CaoRobot::Execute("TRQTIME") コマンド

DRIVE 文のトルク制限指定オプションを使用するときの, 指定された軸に対して電流制限のタイムアウト時間を設定します.

書式　TRQTIME (<lAxis:VT_I4>, <lVal:VT_I4>)

< lAxis >	:	[in] 軸番号 (VT_I4)
< lVal >	:	[in] タイムアウト時間 (VT_I4)
戻り値	:	なし

使用例

```
Dim rbt As Object
Set rbt = caoCtrl.AddRobot("MF20")
rbt.Execute "Trqtime", array(2, 2500)        ' トルク Timeout を 2.5sec
```

付録A. RCXコマンド対応表

Appendix A.1. コントローラクラス

・ Execute メソッド

表 A-1 CaoController::Execute メソッド-RCX コマンド対応表

コマンド名	RCX コマンド
EMGRST	EMGRST
ArithmeticExpression	?”数値式”
StringExpression	?”文字列式”
PointExpression	?”ポイント式”
ShiftExpression	?”シフト式”
SendControlCode	-
NativeSend	-
NativeReceive	-

表 A-2 CaoController の変数オブジェクト-RCX コマンド対応表

変数名	RCX コマンド
@Timeout	-
@CONFIG	?CONFIG
@EXELVL	?EXELVL
@MOD	?MOD
@MSG	?MSG
@UNIT	?UNIT
@VER	?VER
@MEM	?MEM
@EMG	?EMG
@SELFCHK	?SELFCHK
@OPSLOT	?OPSLOT
P	?”ポイント式”
	Pn
S	?”シフト式”
	Sn

Appendix A.2. ロボットクラス

表 A-3 CaoRobot::Execute メソッド-RCX コマンド対応表

コマンド名	RCX コマンド
ABSRST	ABSRST
DRIVE	DRIVE
	DRIVE2
DRIVEI	DRIVEI
	DRIVEI2
ORIGIN	ORIGIN
PMOVE	PMOVE
	PMOVE2
SERVO	SERVO
	SERVO2
HAND	HAND
	HAND2
RIGHTY	RIGHTY
	RIGHTY2
LEFTY	LEFTY
	LEFTY2
SHIFT	SHIFT
	SHIFT2
ARCH	ARCH
	ARCH2
ASPEED	ASPEED
	ASPEED2
AXWGHT	AXWGHT
	AXWGHT2
ORGORD	ORGORD
	ORGORD2
OUTPOS	OUTPOS
	OUTPOS2
PDEF	PDEF
TOLE	TOLE

	TOLE2
WEIGHT	WEIGHT
	WEIGHT2
TORQUE	TORQUE
	TORQUE2
TRQTIME	TRQTIME
	TRQTIME2

表 A-4 CaoRobot::Execute 以外のメソッド-RCX コマンド対応表

メソッド	RCX コマンド
Accelerrate	ACCEL
	ACCEL2
	DECEL
	DECEL2
Change	CHANGE
	CHANGE2
Halt	^C
Move	MOVE
	MOVE2
	MOVEI
	MOVEI2
Speed	SPEED
	SPEED2

表 A-5 CaoRobot の変数オブジェクト-RCX コマンド対応表

変数名	RCX コマンド
@ARM	?ARM
@ORIGIN	?ORIGIN
@ABSRST	?ABSRST
@SERVO	?SERVO
@SPEED	?SPEED
@WHERE	?WHERE
@WHERE2	?WHERE2

@WHRXY	?WHRXY
@WHRXY2	?WHRXY2
@SHIFT	?SHIFT
@HAND	?HAND